

GDDM

This chapter describes how to install the Com-plete/GDDM interface that will enable Com-plete to use color graphics capabilities. Other documentation relevant to the installation of this interface is the *GDDM Release 4 Installation and System Management* (Publication Number SC33-0152)*, hereafter referred to as the *GDDM Installation Guide*. For the installation described in this section, you should be familiar with the GDDM documentation and refer to this section for clarification of the differences in the way GDDM interfaces to Com-plete and CICS. Com-plete and CICS are similar in that they are both teleprocessing monitors. This is the reason that Com-plete uses the same interface to GDDM as CICS. For this installation, then, generally follow the instructions concerning the installation of GDDM under CICS as described in the *GDDM Installation Guide*. This section supplies specific information for the Com-plete environment.

* Current name and number of documentation - these are subject to change.

Having finished the installation process, your installation will have the ability to run the various GDDM editors supplied with the various components of GDDM that you have licensed.

Note:

You need GGDM to install Natural GRAPHICS under Com-plete. See Adding DFHEAI and DFHEAI0 below.

This chapter covers the following topics:

- Installing GDDM
 - Com-plete Components for the GDDM Interface
 - Adding DFHEAI and DFHEAI0
 - VSAM, ADMF, DFHTSD
 - Altering the CICS environment
 - TIBTAB
 - Com-plete JCL Modifications
 - Com-plete Startup Parameters
 - UUTIL/ULIB Activities
 - User Calls
 - Execution
 - Performance Considerations
-

Installing GDDM

The first step is to install GDDM/PGF according to the installation procedures in the *GDDM Installation Guide*. This will result in the creation of four libraries (MVS):

- GDDMLIB non-executable load modules;
- GDDMSAM sample applications, macros, JCL;
- GDDMSYM sample symbol sets;
- GDDMLOAD executable load modules.

or it will create an equivalent set of VSE libraries if VSE is the operating system.

The reentrant code supplied for GDDM/PGF is approximately two megabytes. You must decide where this is to reside, and allocate the space required for it. To help you determine the best place for GDDM, refer to the *GDDM Installation Guide*.

The main consideration when deciding where to place the GDDM code is whether or not there is another subsystem (TSO, CICS, etc.) in your installation that will be using GDDM. If this is the case, you should consider putting GDDM in LPA (MVS) or other common system storage. If there is no other subsystem, it is probably better to place GDDM in Com-plete's region or partition.

Once you have determined where GDDM is to reside, install it according to the procedures in the *GDDM Installation Guide*. In the *GDDM Installation Guide*, you will find a section on installing GDDM with CICS for MVS and information on how to install GDDM with CICS for VSE. Particular attention should be paid to the sections concerning the adding of the CICS command level stubs DFHEAI and DFHEAI0 (see Adding DFHEAI and DFHEAI0 below).

Com-plete Components for the GDDM Interface

The following table describes the Com-plete components of the GDDM interface, and how they are defined within Com-plete. During standard installation all DD names are with default values; however, these may have to be reviewed based on the usage of GDDM and for performance reasons. A sample list of programs which must be residentpaged is contained in member GDDMSAMP on the distributed source data set. This must be reviewed, particularly in relation to new GDDM releases, as new modules may have been included which can also be residentpaged.

Component	Description	Definition
ADMC	The Com-plete application program to invoke the GDDM Interactive Chart Utility.	ULIB CAT
ADMI	The Com-plete application program to invoke the GDDM Image Symbol Set Editor.	ULIB CAT
ADMV	The Com-plete application program to invoke the GDDM Vector Symbol Set Editor.	ULIB CAT
ADMP	The Com-plete application program to invoke the GDDM print utility.	ULIB CAT
TLCIINIT	Com-plete/GDDM initialization processing	RESIDENTPAGE
TLCIMAIN	Com-plete/GDDM control processing	RESIDENTPAGE
TLCISTG	Com-plete/GDDM storage processing	RESIDENTPAGE
TLCIPRGM	Com-plete/GDDM program processing	RESIDENTPAGE
TLCIICP	Com-plete/GDDM "interval control" processing	RESIDENTPAGE
TLCITERM	Com-plete/GDDM terminal I/O processing	RESIDENTPAGE
TLCIFILE	Com-plete/GDDM file I/O processing	RESIDENTPAGE
TLCITRND	Com-plete/GDDM "transient data" processing	RESIDENTPAGE
TLCITEMP	Com-plete/GDDM "temporay stg" processing	RESIDENTPAGE
TLCITRAC	Com-plete/GDDM trace processing	RESIDENTPAGE
TLCIFC	Com-plete/GDDM file processing	
DFHEAI	Com-plete/GDDM interface stub 1. Linked with various mods	
DFHEAI0	Com-plete/GDDM interface stub 2. Linked with various mods	
ADMI	GDDM Symbol Set File DD name	UUTIL FM
DFHTSD	Com-plete/GDDM "Temporary Storage" data set DD name	UUTIL FM

Adding DFHEAI and DFHEAI0

Com-plete supplies you with two modules called DFHEAI and DFHEAI0. These modules are direct replacements for the CICS modules of the same name. Therefore, the section in the *GDDM Installation documentation* entitled **Link-edit with DFHEAI and DFHEAI0** in **Tailoring GDDM step-by-step for CICS** must be repeated or initially done using the Com-plete replacement modules.

Take the sample job stream described in this section for linking GDDM with DFHEAI and DFHEAI0 and change the loadlib from which the modules are linked to point at the Com-plete load library. This will cause the Com-plete replacement modules to be included.

Be careful, however, that the various GDDM modules have not already been linked with the CICS modules, as a second include will simply make the module bigger and use the previously included modules. Example job streams are also provided for GDDM-PGF, GDDM-IMD and GDDM-IVU in the source library. Repeat the same procedure for each of these if this is applicable for your installation. If a sample GDDM job is not available, a sample job called GDDMLINK is supplied on the distributed source data set.

Please note that Natural GRAPHICS uses the system programmer's interface (module ADMASPC) to GDDM, which requires the modules DFHEAI and DFHEAI0. These modules must be taken from the Com-plete load library.

VSAM, ADMF, DFHTSD

Prepare your VSAM files as stated in the *GDDM Installation Guide*. You can use ADMF and other similar VSAM files under Com-plete, just as the function GDDM ESLIB describes.

You must create an additional VSAM file required by Com-plete's GDDM Interface. Its DDN to Com-plete must be DFHTSD. A sample MVS job stream to create it follows below:

```
//jobname JOB (JOB card information)
//JOB CAT DD DSN=USERCAT,DISP=SHR
//ALLOCEXEC PGM=IDCAMS
//dd1DD UNIT=SYSDA,VOL=SER=VOLSER,DISP=SHR
//SYS PRINT DD SYSOUT=A
//SYS IN DD *
        DEFINE CLUSTER -
            (NAME(index.DFHTSD) FILE(dd1) -
            UNIQUE -
            VOLUMES(volser) -
            SHAREOPTIONS(3 3) -
            RECSZ(400 3000) -
            CONTROLINTERVALSIZE(4096) -
            KEYS(10 0)) -
        DATA -
            (RECORDS(300 100)) -
        CATALOG (USERCAT)
/*
```

After defining this file, it must be initialized. A sample job stream to initialize it would be:

```
//jobname JOB (JOB card information)
//VSAM EXEC PGM=IDCAMS
//SYS PRINT DD SYSOUT=A
//DD1DD DSN=index.DFHTSD,DISP=SHR
//STEP CAT DD DSN=USERCAT,DISP=SHR
//DATA DD *
FIRSTONE THIS IS THE FIRST RECORD TO CREATE IT
/*
//DD3 DD UNIT=SYSDA,DISP=OLD,VOL=SER=VOLSER
//SYS IN DD *
        REPRO INFILE(DATA) -
        OUTFILE(DD1)
/

// JOB GDDMTSDDEFINE DFHTSD
// DLBL UOUT,'FIRST.RECORD',0,SD
// EXTENT SYS005,xxxxxx,1,0,nnnnn,1<=====
// ASSGN SYS005,DISK,VOL=xxxxxx,SHR <=====
```

```
// ASSGN SYS004,SYSIPT
// EXEC OBJMAINT
./ CARD DLM=
./ COPY
FIRSTONE THIS IS THE FIRST RECORD TO CREATE IT
/*
// OPTION LOG
// DLBL IJSYSUC,'vsamusercatalog',,VSAM<=====
// DLBL DFHTSD,'GDDM.DFHTSD',,VSAM <=====
// ASSGN SYS007,DISK,VOL=xxxxxxx,SHR<=====
// DLBL INFILE,'FIRST.RECORD',0,SD
// EXTENT SYS007,xxxxxxx
// DLBL DFHTSD,'GDDM.DFHTSD',,VSAM
// EXEC IDCAMS,SIZE=AUTO
    DEFINE CLUSTER -
        (NAME(GDDM.DFHTSD) -
         VOLUMES(xxxxxxx) -
         SHAREOPTIONS(3 3) -
         RECSZ(400 3000) -
         CONTROLINTERVALSIZE(4096) -
         KEYS(10 0)) -
        DATA -
        (RECORDS(300 100)) -
        CATALOG (vsamusercatalog)<=====
    REPRO INFILE(INFILE ENV(BLSZ(80) RECFM(FIXUNB) RECSZ(80))) -
        OUTFILE(DFHTSD)
/*
/&
```

All VSAM files must be defined to Com-plete using the FM function of online utility UUTIL. If you specify the files' data set names there, there is no need to include the DD statements in the Com-plete JCL.

Altering the CICS environment

The headings in the *GDDM Installation Guide* regarding the modification of CICS tables is of no interest to users of GDDM under Com-plete. CICS uses a series of tables to keep track of the files it uses and the destinations for certain terminal requests. Com-plete uses several different mechanisms for the same functions. For instance, Com-plete users catalog the VSAM files they use online with UUTIL FM. You can therefore ignore the sections in the GDDM documentation which describe the altering of CICS tables. The only CICS-like table Com-plete maintains is a file control table which is referenced internally by the Com-plete/GDDM interface. If you wish to define additional VSAM files for use with symbol sets, Com-plete will dynamically add entries into this internal table, thus freeing you from such table maintenance.

TIBTAB

The philosophy behind this is to reduce the amount of maintenance for the system programmer to an absolute minimum. Therefore, rather than having to define the attributes of different devices in different places (eg. LOGMODE and CICS TCT), Com-plete takes the LOGMODE for a device to indicate exactly what that device is.

Therefore, once all the device types are defined in the COMPLETE TIBTAB, when a terminal or its capabilities change, the only change necessary is an update of the LOGMODE for the device. Care must be taken with logmodes for devices to ensure that they indicate what the device can actually do. In this

case, the relevant indicator is an indicator that the device can support a Read-Partitioned-Query command. Once this is possible, the software can work out the rest. IBM supply various examples of LOGMODES for various device types.

In general, VDUs need not be specified in the TIBTAB as they can be allocated dynamically. Printers can also be allocated dynamically. However, it is sometimes better to define them in the TIBTAB to avoid having to reallocate them after a restart of Com-plete, unless a fully dynamic TIBTAB is specified (via TIBTAB=DYNnnnnn in the Com-plete sysparms). A sample TIBTAB follows. This defines two printers PRT1 and PRT2. A maximum of 100 terminals are available with this TIBTAB.

```
TIBTAB    TIBSTART NOTIBS=100
PRT1      TIB      1,VTAM,3287L,NAME=PRT1,OPT=(ACQUIRE,SHARE)
PRT2      TIB      2,VTAM,3287L,NAME=PRT2,OPT=(ACQUIRE,SHARE)
          TIBEND
```

Note:

Terminals defined in the TIBTAB can be changed if information given in the LOGMODE conflicts with that specified in the TIBTAB.

Com-plete JCL Modifications

Com-plete must be able to find the various GDDM and Com-plete/GDDM interface modules. To make the GDDM modules available, the GDDM load library must be included in the Com-plete COMPLIB concatenation AFTER the library where the newly linked CICS/Com-plete-dependent modules reside (that is, the modules that are linked with DFHEAI and DFHEAI0).

To make the Com-plete/GDDM interface modules available, the Com-plete distributed load library must also be in the COMPLIB concatenation. However, you have the option of copying the relevant modules to a library which is already in the concatenation if the inclusion of the Com-plete distributed load library in the COMPLIB concatenation is not desirable.

Com-plete Startup Parameters

GDDM/PGF requires a considerable amount of work area. This will be allocated in the thread. Load as much of the code as possible outside of the thread to allow more room for the work area. The size of the thread to allocate will depend on the amount and type of work done. In a lot of cases this will be trial and error, so you are advised to start with the Software AG supplied default listed below. You must therefore review thread sizes to ensure that at least one is available that is big enough to run the various programs.

To reduce thread and catalog sizes, as much of the GDDM nucleus and code as possible should reside outside the thread. If GDDM is installed in the LPA, Com-plete can use these modules. However, if this is not the case, Com-plete can load them into its own nucleus via the RESIDENTPAGE sysparm. All eligible residentpage modules should be included using this parameter. A module is eligible when it is Reentrant and Reusable which, in fact, a large part of the GDDM nucleus is. The best thing to do is to look at the modules required to be defined in the CICS PPT. Take this list and delete any module not residentpage-eligible. The result is a list of modules which you must include in the Com-plete nucleus via the RESIDENTPAGE sysparm. A sample set of RESIDENTPAGE parameters is supplied in the GDDMSAMP member of the distributed source data set.

To enable the Com-plete VSAM interface to function, the VSAMDS sysparm must be specified. A starting value for this could be VSAMDS=2 indicating that a maximum of two VSAM files will be used for the duration of the Com-plete session. This is the current minimum that the Com-plete/GDDM interface requires to function. This can be increased as the need for more VSAM files arise. Please see the subsection **Performance Considerations** below concerning more VSAM parameters.

UUTIL/ULIB Activities

This subsection describes the various default parameters for the GDDM/Com-plete components for UUTIL FM and ULIB. If you have used the standard Com-plete installation procedure, these entries already exist. However, you may wish to change them for performance reasons.

Define the VSAM files to Com-plete using the FM subfunction of UUTIL. You must specify the following options:

For ADMF and DFHTSD:	Retrieval Update Add STRNO=2 MACRF=(DIR,NSR,NDF,SEQ)
----------------------	--

The following are the ULIB CAT parameters for the programs:

ADMI	IMAGE SYMBOL SET EDITOR	768K
ADMP	printer utility	768k
ADMV	vector symbol set editor	768k
ADMC	chart utility	768k
TLCIFC	VSAM file control table	(no size need be specified)

User Calls

The WRTSF function of Com-plete is available to users of graphics. See the Com-plete Application Programming documentation for details. It is strictly for use with extended function 3270's:

```
CALL WRTSF|D|C|R|      to perform a write structured field
                        to a specified TIB.
```

This CALL has the same structure as all the other writes. It is for performing write structured field commands. If you wish to perform a Read-Partition-Query, since this generates an input interrupt, use the WRTSFC option. You will get the results of this read in the input buffer. Be sure to use the READS form of the read CALL.

For loading symbol sets, use the WRTSFR option, as you do not need to wait for any input to occur. Alternatively, you can load programmed symbols using the UMAP utility LOAD PROGRAMMED SYMBOLS function (see the Com-plete Application Programming documentation).

The WRTSFD option is used when you want to set something and exit. For more information about the Write Structured Field command, consult the *IBM 3270 Information Display System documentation* (Publication Number GA23-0060) and the *3274 Control Unit Description and Programmer's Guide* (publication number GA23-0061).

When you use this form of the WRITE, Com-plete does NOT check for data to be output. The user program must ensure that the data being sent is valid for the WRF terminal command.

Execution

When executing GDDM/PGF in the Com-plete environment, use the CICS instructions in the GDDM and PGF documentations. Some differences exist, and these are discussed below.

CICS abends with alpha abend codes, that is, GDDM will ask for an abend G000. Com-plete converts that to abend 7000. If GDDM has issued an abend code that is alpha, Com-plete changes the zones to "F" and continues as though it was numeric.

Com-plete's interface abend codes are:

922	function not supported.
923	unrecoverable terminal write error.
924	no storage available in the thread
925	error in the ask time routine
926	error in ENQ or DEQ
927	error in loading or releasing programs.
928	error in initializing the graphics environment.

Performance Considerations

GDDM is resource-intensive, but some tuning can be done. Firstly, review the CPUTIME and REALTIME sysparms for Com-plete. These will avoid superfluous abends and messages if they are specified correctly.

GDDM makes quite heavy use of its VSAM files and therefore allocation of buffers to these data sets should be carefully reviewed with the expected usage in mind. VSAM itself handles buffer usage and therefore the only control you have is to allocate more or less buffers. This can be done on an individual file basis or the files can be grouped into a VSAM Local Share Resources pool (LSR). This is the most efficient in that all buffers are available for use for all files (given the restrictions of VSAM LSR, see the relevant VSAM documentation) as opposed to individual buffers which use storage even when the file is not in use. See the section **Using VSAM with Com-plete** and the Com-plete sysparms VSAMBUFFERS and VSAMRPL.